

VND

Luís Felipe Ramos Ferreira

Como heurística VND (*Variable Neighbourhood Descent*) foi desenvolvido o algoritmo de *2-OPT*. O algoritmo tem como ideia principal pegar duas rotas no caminho do caixeiro que passam uma em cima da outra e reordenar o caminho para que isso não ocorra. Isso é bom pois, dada a desigualdade triangular das instâncias utilizadas para o TSP, a nova rota encontrada será com certeza melhor. No entanto, o algoritmo apesar de eficiente e bom não retorna uma solução ótima, uma vez que podemos ficar presos em um ótimo local durante sua execução. Essa abordagem é usada em diversos problemas de natureza combinatória difícil, como no caso do TSP e também para problemas como VRP (*Vehicle Routing Problem*). O código com a implementação dessa heurística pode ser encontrado neste [repositório](#) no arquivo *vnd.py*.

Algorithm	Instance	Path Weight	Time (s)
2-OPT	kroD100.tsp	22922	6.09e-08
2-OPT	lin105.tsp	15780	5.99e-08
2-OPT	att48.tsp	10933	5.51e-08
2-OPT	kroB150.tsp	28553	5.79e-08
2-OPT	berlin52.tsp	8352	6.08e-08
2-OPT	kroA200.tsp	32336	6.48e-08
2-OPT	rat195.tsp	2547	7.96e-08
2-OPT	pr152.tsp	77591	7.19e-08
2-OPT	st70.tsp	720	6.10e-08
2-OPT	pr144.tsp	61507	8.28e-08
2-OPT	pr124.tsp	60981	5.86e-08
2-OPT	pr136.tsp	103206	6.17e-08
2-OPT	kroB200.tsp	33576	7.14e-08
2-OPT	kroB100.tsp	24324	5.90e-08
2-OPT	kroA150.tsp	30500	6.28e-08
2-OPT	kroA100.tsp	23675	5.87e-08
2-OPT	pr107.tsp	45099	5.70e-08
2-OPT	kroE100.tsp	24408	5.50e-08
2-OPT	rat99.tsp	1292	5.44e-08
2-OPT	kroC100.tsp	22256	5.64e-08
2-OPT	pr76.tsp	125277	9.05e-08

Table 1: Resultado médio para as execuções